Journal of Medical Robotics Research, Vol. 0, No. 0 (2015) 1–12
♥World Scientific Publishing Company

Safe Human-Robot Interaction in Medical Robotics: A case study on Robotic Fracture Surgery System

Ioannis Georgilas, Giulio Dagnino, and Sanja Dogramadzi

Bristol Robotics Laboratory, University of the West of England and University of Bristol, Bristol, BS161QY, United Kingdom
E-mail: ioannis.georgilas@uwe.ac.uk

This paper presents a safety analysis of a Robotic Fracture Surgery System using the Systems-Theoretic Process Analysis (STPA). It focuses particularly on hazards caused by the human in the loop. The robotic system and operating staff are modelled including information flow between different components of the system. The analysis has generated a set of requirements for the system design that can ultimately mitigate the identified hazards, as well as a preliminary set of human-factors that can improve safety.

Keywords: Safety, Medical Robotics, Surgical Robotics, human-robot interaction

1. Introduction

Safety in close human-robot interactions has been a topic of many recent research projects. Use of medical robots in operating theatres brings up concerns on every system level – individual components, system and human-robot interaction. Even though medical robots can potentially provide significant benefits to patients, their operation can be significantly more harmful if not safely utilized.

A recent report by Alemzadeh et al on a 13-year-long study of Da Vinci system's FDA data [1] assesses safety of Da Vinci system based on the recorded faults that include 4,798 adverse events (involving 86 deaths, 410 patient injuries, and 3405 device malfunctions). A thorough data processing shown that the reported accidents occur due to the 'inadequacy of safety controls and comprehensive warnings to the surgeon, limited safety and training practices, lack of certification, and limited surgical experience'. A similar study, conducted on computer-based medical devices, reports the device software as the major cause for recalls (64%) followed by hardware and I/O module failures. The same study emphasises the importance of design with well-defined safety requirements and robust error-detection methods.

In medical robotics (MR), the surgeon and clinical staff are often required to interact with the system that can have a wide range of engineering complexities and requirements for the human input. Human-robot interaction plays an important role in surgical safety and has been largely ignored in safety analysis of medical robots. Effects that human cognitive and emotional state can have on tasks jointly performed with a robot has already been reported for a search and rescue scenario [2]. It can be argued that surgical environment can be sufficiently similar, putting operating theatre staff under strain while requiring interaction and operation of a complex medical robot.

1.1. *RAFS project*

Bristol Robotics Laboratory has developed a new robot-assisted system for minimally invasive treatment of joint fractures - Robot Assisted Fracture Surgery (RAFS) system [3], [4].

RAFS aim is the anatomical reduction of intra-articular fractures with pre-surgical planning carried out by the orthopaedic surgeon. The robotic system and the surgeon have to undertake a series of safety-critical actions that are part of a fracture management surgical flow. This imposes two safety-critical objectives of the system in order to achieve an accurate anatomical reduction of the fracture –

- Virtual reduction in the pre-planning stage by the surgeon and
- Intra-operative physical reduction performed by the robotic system;

1.2. *Safety Analysis*

With intensive research and widespread commercial presence of medical robots, safety concerns are becoming increasingly important to resolve.

Several safety frameworks have been so far proposed in literature to tackle potential hazards imposed by complex medical robotics systems [5]–[7]. A recent example is based on the component based software engineering [8]–[10] that decomposes MR safety features into a run-time software platform. The platform monitors the components of the system and is intended to be reusable and deployable to any MR system. This concept is based on the run-time software architecture that is reconfigurable and middleware independent. However, this layered architecture and its safety libraries do not include human-generated errors. In a demanding and stressful surgical environment, this kind of errors typically occurs in human-robot and human-computer interactions.

With the current work we are investigating interactions and how they are affecting the safety of the system. Specifically, we are investigating a new methodology for identifying dangerous situations and design appropriate safety measures. This document is structured in the following way. In section 2 a description for safety analysis approaches is given, and the steps for the implementation of STPA is described. In section 3, the RAFS system, its individual parts, and the clinical workflow are presented. In section 4 the implementation of STPA for the RAFS system is performed. Finally, in section 5 the results from the safety analysis are reported and an investigation on how the results can be utilised to improve the safety of the system done.

2. Safety Analysis Methods

The existing risk analysis tools - Failure Mode and Effects Analysis (FMEA), Functional Hazard Analysis (FHA), HAzard and OPerability study (HAZOP) have been developed in the last 50 years for standard engineering systems with components that required relatively basic interaction with the user. These tools are effective in identifying direct relationships between causes and faults but often-indirect dependencies may not be easily recognized. Moreover, independence may be assumed to exist when it does not.

Most commonly utilised risk-assessment tools are approaching complex issues as chains-of-failure-events with a prevalent method of prevention being safeguarding against cascading failure from one component to the next. This is typically achieved by monitoring a component of the system, detecting its failure, reacting on it, and recovering before it propagates to the next level. For most electromechanical parts the two first steps of this process are relying on the component reliability that can be exhaustively tested, and a probabilistic assessment of the risk they pose. Robotic technologies are based on increasingly complex system designs that no longer can be exhaustively tested using the current methods applied to complex systems such as the ones in e.g. aerospace industry. Utilization of robots in dynamic human environments poses new problems and this type of risk-assessment oversimplifies user behaviour e.g. if he/she gives a correct/incorrect command. This does not consider the fact that the modern system decision-making required by users can be a complex cognitive task. User-generated faults cannot be assessed in isolation but instead considered within the specific context of the system.

A preliminary hazard analysis of RAFS system has been performed for a range of user-system interactions. We used Universal Modelling Language (UML) and codification defined in a Hazard-Effect Table. In order to elaborate the importance of hazard identification we focused on two specific examples, one from the pre-operative planning and one from intra-operative interactions. For the former we selected the virtual reduction scenario and analysed that virtually negligible errors in the interaction can become serious problems further down the process [11].

This work has been expanded to obtain a deeper understanding of interactions and causalities that can impede on the system safety.

2.1. *STPA Analysis*

STPA [12] or Systems-Theoretic Process Analysis is a hazard analysis technique used to identify potential hazard scenarios and mitigation methods to prevent accidents and failures. STPA is derived from system theory unlike most other hazard analysis tools that are derived from reliability theory. This system approach allows a wider overview of the process in question, and a hierarchically structured analysis that has already been explored for domestic robot hazard analysis [13] and medical applications [14].

Another benefit of the STPA analysis is the ability to take into consideration models of each 'actor' in the system operation. Actor models in this context are part of the structure and state of the system's control architecture. The term actor could refer to e.g humans-in-the-loop or the robot controller.

In order to use the STPA approach to analyse hazards of a system, a sequence of steps [15] must be utilized to facilitate the analysis, namely:

(i) Establish the system's Hierarchical Control Model (HCM) and Process Model (PM). In this step the hierarchy of the control elements of the system is developed, moreover, the interconnection of the elements and the signals exchanged as part of the process are identified;
(ii) Identify potentially Unsafe Control Actions (UCA). In this step, an analysis of the effects of a control action, and the safety implication within specific context is performed;
(iii) Use UCAs to derive safety requirements and constraints (SR&C). In this step, the context conditions upon which a UCA can materialise are translated into constrains for the control system and requirements of the design.

The humans-in-the-loop have a cognitive model based on their role in the system operation. The robot controller also has a model of the system's control process and the relevant sensory information. All actors issue control and setup commands based on these models. Any discrepancies between reality and model might lead to hazardous situations. In other words, a safe control state might become an unsafe due to the conflicting control messages being sent to the system and the context in which these messages are issued.

This holistic approach of STPA makes it ideal for systems where users' input holds significant weight on the overall safety of the system. We have implemented this approach to the RAFS system where complex interactions are directly affecting the performance and quality of surgical outcomes. The work presented here is a system-level analysis of the human-in-the-loop related risks related to medical/surgical robots. The main assumption is that the individual component level risks are known and the focus is on analysis of the complex control

interactions between the actors involved in surgical tasks. The tasks presented in this paper are workflow steps of robot-assisted fracture surgeries but can be translated to other similar procedures. As stated in [15] "The goal is not to find just failures or inadequate operation of individual components in the control loop, but to identify scenarios and combinations of problems that could lead to unsafe control."

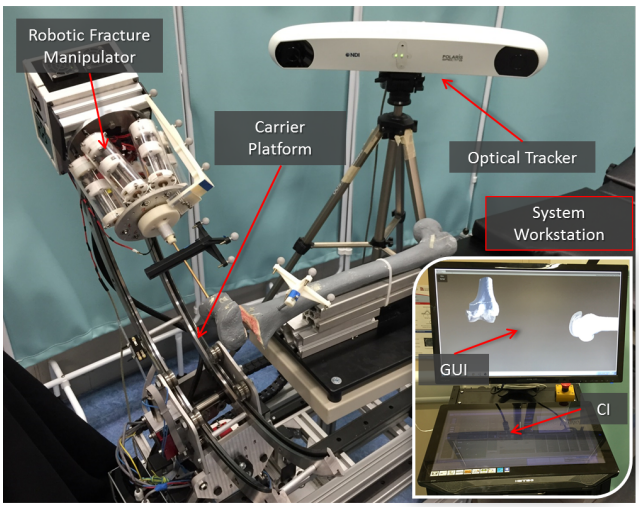


Fig.1 – The RAFS system prototype

3. RAFS System Description

The RAFS system has been developed at Bristol Robotics Laboratory. The RAFS system is shown in Fig. 1. Its subsystems are:

Robotic Fracture Manipulator (RFM) introduced in [16], is designed to be connected to the bone fragment through an orthopaedic pin for fragment manipulation. It is based on a parallel-robot configuration with 6-DOF and has 6 motorized linear actuators fully computer-controlled. RFM's positioning accuracy is *(±10.25mm* along x, y, *±15mm* along z and rotational limits of *±17°* around each axis). Its overall translational accuracy is 0.03±*0.01mm* and rotational accuracy is *0.12±0.01°* [17]. A 6-DOF force-torque load cell attached to RFM enables force control. In order to fully cover the required operational workspace (see [17]), the robotic manipulator is mounted on a carrier platform.

Carrier Platform (CP) is used for a coarse positioning of the RFM (which is rigidly connected) close to the orthopaedic pin. , The RFM is then used to accurately manipulate the fragment to the desired pose. The CP has 4-DOF, two prismatic and two revolute (see Fig.4a), and a cylindrical workspace (700mm length, 300mm diameter), covering the required operational workspace described in [17]. The CP has 4 motorized actuators, one for each DOF, and it is fully computer-controlled.

System Workstation employs a host-target structure composed by a PC (host) and a real-time controller with FPGA (target), and a low-level motor controller. The host PC runs the Graphical User Interface (GUI) and the Configuration Interface (CI). It creates the link between the surgical team and the robotic system. The GUI allows the surgeon to interact with the virtual surgical field (3D Imaging System) while the CI is used for system configuration and safety alarm messages. The host PC communicates with the target controller via Ethernet. The target controller (NI-compactRIO 9068, National Instruments) process users' commands and sends the motion commands to the low level motor controller (EPOS 2 24/3, Maxon Motor) that executes the movement of the robotic system.

3D Imaging System, introduced in [18], [19], consists of a reduction software, an optical tracking system, and a user controller. The reduction software receives pre-operative CT scan data of the fracture and generates corresponding 3D models of the bone fragments. The GUI displays the 3D models and facilitates intra-operative planning of fracture reduction, i.e. virtual reduction. Collision avoidance is enabled to avoid overlap of the 3D models. The optical tracking system (Polaris Spectra, NDI Inc.) provides real-time (25Hz) update of optical tools (*0.25mm* accuracy) connected to the bone fragments and the RFM.

In Fig. 2, the interaction diagram between the subsystems and the human actors is given. The four subsystems are depicted - the vision tracker, the console PC, the system controller and the robot in the surgical field. The human actors are also presented, with the interactions of the surgeon and the system components. For clarity, the actual interactions of the operating theatre (OR) staff have been omitted, but this will be analysed later in the document.

3.1. *Workflow*

Every medical procedure must have a specific workflow that has been scrutinized in terms of its safety. While developing the RAFS system, special effort was made to keep the workflow similar to the current practice and reuse the workflow elements (e.g. placement of manipulation pins). Fig. 3 shows the current surgical workflow of a distal femoral fracture that has been created after discussions with surgeons and evaluated and amended through laboratory experiments.

From Figs. 2 and 3, it is clear that the surgeon is a critical part of the control loop. He/she is providing the expert knowledge and instructs the robotic system to perform the operation; during the pre-operative phase by performing the virtual reduction, and intra-operatively by inserting the pins, supervising and continually assessing the procedure. The surgeon is also responsible for activating the emergency system when the procedure deviates from the pre-planned parameters. Also, the OR staff are performing tasks related to and close to the system such as positioning the tracker, cleaning tools for better visibility, and checking patient and equipment status. All actions

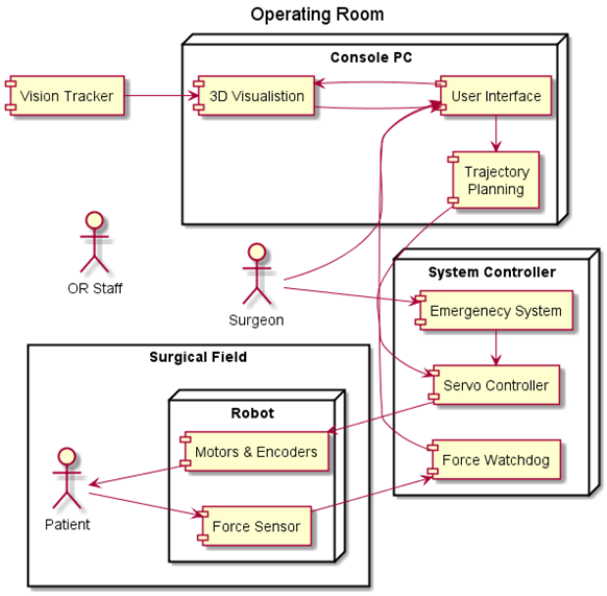


Fig. 2 – Interactions between the subsystems (Vision tracker, Console PC, System Controller, Robot) and human actors (Patient, Surgeon, OR Staff).

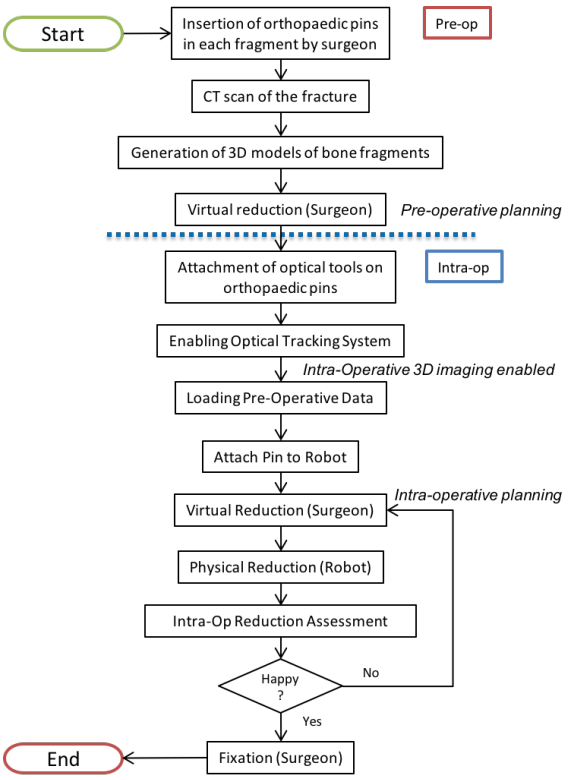


Fig. 3 – RAFS Clinical Workflow. The process is separated into Pre-operative and Intra-operative states.

of the surgeon and staff are performed based on real time situational awareness, clinical experience, and their cognitive and emotional status.

4. STPA for RAFS

The steps, defined in Section 2.1, are applied to setup an STPA analytical process. First, a model of control interactions between the human operators and the robotic system is developed, then control actions of both the human and the robot are identified, and, finally, constraints and requirements to be implemented in the robot and the surgical workflow are defined.
The STPA definitions [15] for accidents and hazards are:

Definition 1. "An accident is an undesired and unplanned event that results in a loss, including loss of human life or human injury, property damage, environmental pollution, mission loss, financial loss, etc."

Definition 2. "A hazard is a system state or set of conditions that together with a worst-case set of environmental conditions, will lead to an accident (loss)."

Based on these definitions, and clinical analysis of the workflow, the following major potential accidents, and their respective types of consequences are:

AC1 Person Injury (Patient or OR staff);
AC2 Inaccurate Reduction – Mission Failure;
AC3 Tissue Damage – Mission Failure.

The hazards that can cause these accidents are the following:

HA1 Uncontrolled motion of the System;
HA2 Incorrect motion of the System;
HA3 Incorrect setup of the System.

The first hazard can be further analysed into more specific ones, like uncontrolled motion of the system when attached to the fragment and/or when moving to reach the fragment's final position.

From this analysis we can further refine high-level requirements identified for RAFS in section 1:

- Must accurately reduce an intra-articular fracture;
- Must not move in an unintended way;
- Must not lose position and force feedback.

Table I – Table of Hazards and Corresponding Accidents

Hazards	***Uncontrolled motion of the System when …***		***Incorrect motion***	***Incorrect Setup***
Accidents	***… attached to fragment***	***… moving to new position.***		
Human Injury	*x*	*x*	*x*	*x*
Inaccurate Reduction	*x*		*x*	*x*

Soft Tissue Damage	*x*		*x*	*x*

The next step of the STPA analysis is to map the control structure of RAFS into the HCM and PM. Based on this and the process model that each member of the hierarchy is represented by, a list of potentially unsafe control actions will be derived.

4.1. *Hierarchical Control Model (HCM) and Process Model (PM)*

The RAFS system is a set of sub-systems and interactions, as shown in Fig. 4. The RAFS's HCM is represented by three basic elements and command and information signals exchanged in the system.

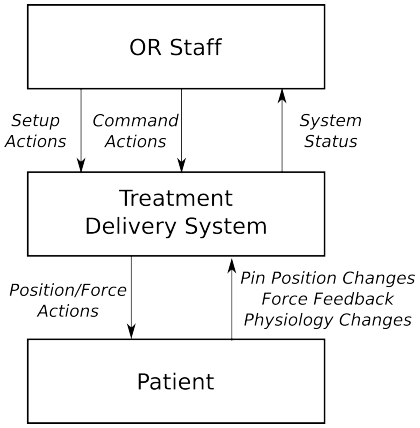


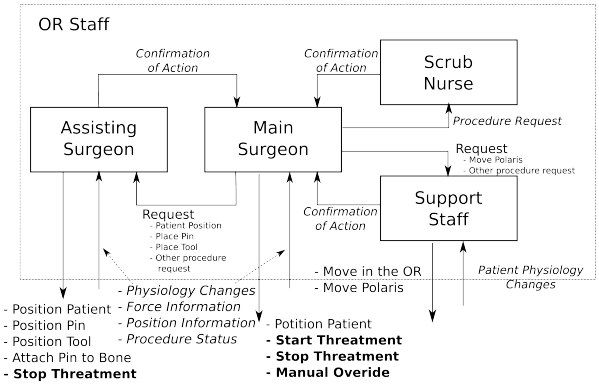
Fig. 4. – Top-level Hierarchical Control Model of RAFS. The three main elements and the types of the exchanged information are presented.

Figure 5 shows a detailed information flow between OR Staff and the Treatment Delivery System and the Treatment Delivery System and the Patient. Both Figs 5a and 5b are focused on the intra-operative phase. Three types of information signals are defined:

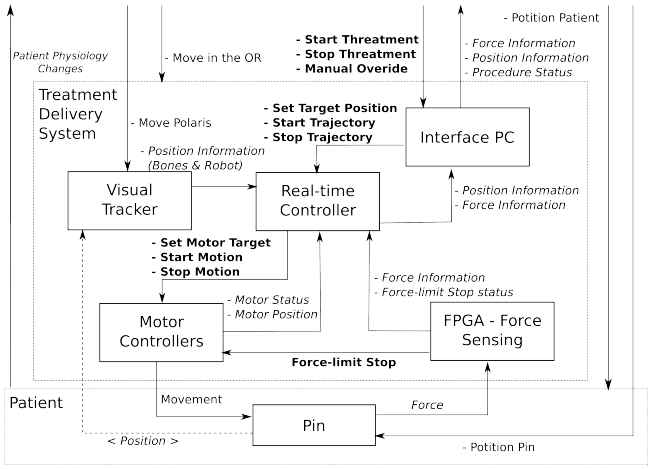
- Command Actions – **bold** text in Fig. 5;
- Setup Actions – plain text in Fig. 5;
- Feedback information – *italic* text in Fig. 5.

Notably, the control actions are divided into two major groups, setup and command, having roughly equal importance for the safe operation of the system.

An important aspect of the STPA is an accurate process model of each actor in the system. Discrepancies between the model and the reality are typically leading to hazardous situations. The process models in the RAFS system are presented in Fig. 6.



(a) Control model for OR staff



(b) Control model for Treatment Delivery System and Patient

Fig. 5. – Detailed views of the Hierarchical Control Model. Command signals are in **bold**, setup signals in plain, and information in *italic*. The *position* information from Pin to Visual Tracker (subfigure b) is indirect.

The OR staff has two models, a model of the clinical procedure and a model of the treatment delivery system. The former is associated with the surgeon's and the OR staff's cognitive model of the procedure based on their experience and hospital procedures. These models include setup characteristics and anticipated behaviour based on training on the system and familiarity with its operation.

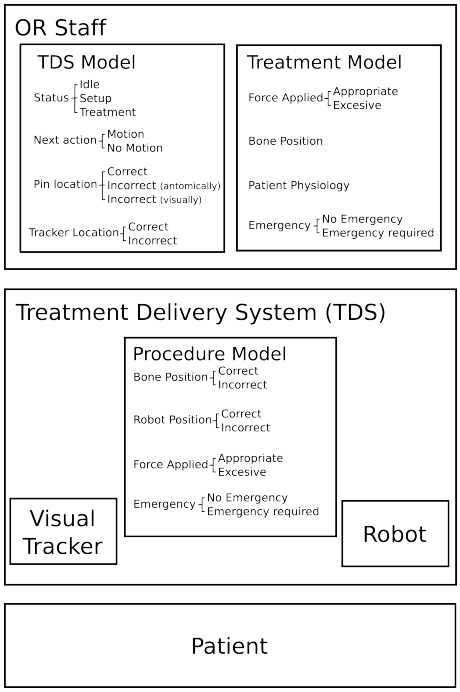


Fig. 6. – Process Models for the Sub-Systems of RAFS

After the HCM and the PM of the system have been established the next step for the implementation of STPA is to find the Unsafe Control Actions (UCA). The UCAs are usually generated by the HCM and must be tracked back in the PM.

4.2. *Identifying Unsafe Control Actions (UCA) in RAFS*

Based on the HCM (Fig. 5) presented above we define the Unsafe Control Actions (UCA) for the system. In this work we are going to extend the definition of UCA to include setup actions. As explained above, the reason for this is that the setup actions are crucial for the system safety. A table representation of the UCAs includes not only the hazards associated with UCA, but also wrong timing and out-of-sequence control actions (CA) as too early or late use of the CAs. The UCAs for RAFS are given in Table II. The CAs and hazard situations depend on the time of issuing the command. If a hazard does not occur under any circumstances, then the action is safe in the given time condition.

For some UCAs the context of the action is also required in order to confirm if a hazard is probable. One example of this is CA2-Stop Treatment. When issued too early, it can cause the robot not to complete the bone reduction. This depends on the current state of the system, which the actor has to evaluate based on his/her observation of the surgical state. An STPA tool to analyse possible system states is the use of context tables. In these, the state of the system is correlated with the process parameters to establish if an action is hazardous in a given context or not. Table III provides this information for the UCA specified in Table II. Setup Actions are omitted because these are affecting the state of the system and also the state of the process models.

In the following section we are going to investigate the impact of setup actions to the process models and corresponding hazards.

4.3. *The effects of setup actions*

The setup actions are affecting directly the state of the process model both for the OR stuff and for the TDS. As can be seen in Table II, all the time critical criteria for a hazard are related to the pin, bone or visual feedback position. The potential compromise each Setup action (SCA) can have to a process model parameter (from Fig. 6) leading to one or more potential hazards is given by the following statements:

Position Patient **compromises** *Bone Position/Tracker Location* **leading to hazards** *HA2/HA3.*

Position Pin **compromises** *Pin Location/Tracker Location/Bone Location* **leading to hazards** *HA1/HA2/HA3.*

Position Tool **compromises** *Tracker Location/Bone Location* **leading to hazard** *HA2.*

Move in the OR **compromises** *Tracker Location* **leading to hazard** *HA2.*

Move Polaris **compromises** *Tracker Location* **leading to hazards** *HA3/HA2.*

Attach Pin to Robot **compromises** *Bone Location/Force Applied* **leading to hazard** *HA2*

The next step after establishing the context of the control action is to investigate control scenarios that involve those actions. Based on these scenarios, a set of requirements for the system design can be derived.

5. Discussion

From the analysis presented in the previous section, requirements and constraints for the system that should minimise the identified hazards and thus accidents can be derived. The requirements and constraints will depend on the type of the action, setup or control. The requirements will ensure that the context for a given control action is appropriate, but a more complex checking mechanism must be in place to confirm the state of the process model and ensure that model parameters and the real-time values are synchronised at all times.

5.1. *Requirements and Constrains from UCAs*

Scenarios for the issuing of control actions are derived from Table III. As we note from the table, the most crucial process parameter is the state of Emergency.

CA2 Case 1 – The 'stop treatment' command must not be issued late in the state of emergency to avoid human injury (OR staff or patient). This can be achieved by providing the TDS with emergency parameter information, and ensuring that the operating surgeon has a clear indication of the emergency state.

CA2 Case 2 – The stop treatment command is issued at the appropriate time when the force is excessive. Too early or too late issuing will result in inaccurate reduction or soft tissue damage. This can be achieved by allowing appropriate force limit thresholds and time of reaction, i.e. not an absolute cut-out value but gradient force information. An appropriate representation of force information should be available to the surgeon.

CA2 Case 3 – The stop treatment command is issued when the bone position is not correct, resulting in inaccurate reduction. This can be prevented by clearly indicating that the reduction is not complete and the user interface restricts the issue of this command.

CA2 Case 4 – The stop treatment command is issued when the robot is in an incorrect position, having a mechanical fault, but the reduction process is correct (bone is in a correct position). This can lead to a patient injury if the command is issued too late. An amendment action to prevent this hazard would be that the system continuously evaluates the progress of the treatment in terms of the bone location and stops any other movements.

CA3 Case 1 – The manual override command is issued at the state of emergency. This might lead to patient's injury. There are two requirements/constraints that can amend this scenario; firstly the process model of the TDS must be aware of the emergency state, as in CA2 Case 1, so as to ignore the manual override, secondly, there must be a clear indication of this emergency state to the operating surgeon issued by the TDS.

CA3 Case 2 – The manual override command is issued when an excessive force is applied by the system. This might happen when the system stopped the motion due to reaching the force limits. The user creates this action. This case can be amended by explicitly denying the motion if the force constraints are not reached and informing the user that this is the case.

CA3 Case 3 – The manual override command is issued when the bone is in the correct position. A potential hazard could be inaccurate reduction. A requirement to amend this is to prevent issuing commands when the reduction is completed.

TA2/5 Case 1 – The TDS issues a start trajectory/motion command when the system is in the emergency state, leading to patient's injury. The potential amendment of this case is similar to CA3 Case 1 -the TDS process model is aware of the emergency state of the system.

TA2/5 Case 2 – The TDS issues a start trajectory/motion command when the system is in Setup state and the OR staff is executing setup actions. This will lead to two potential accidents; firstly the OR staff might get injured, secondly the process model is compromised since the movement is affecting the setup process, leading to inaccurate reduction. The safety constraint for this can be to prevent any movement command while in the setup mode.

TA2/5 Case 3 – The TDS issues a start trajectory/motion command when excessive force is applied, leading to soft tissue damage. The requirement in this case is using force threshold as a watchdog and preventing any motion when force limits are exceeded.

TA2/5 Case 4 – The TDS issues a start trajectory/motion command when the bone is reduced but there is a potentially incorrect position of the robot. This is similar to CA3 Case 3 and has the same requirement where the system is to stop issuing commands when the reduction is completed.

TA3/6 Case 1 – The TDS issues a stop trajectory/motion command late when in an emergency situation. Similarly, to other cases (CA2 Case 1), the system should be able to recognise the emergency status and handle stop commands accordingly.

TA3/6 Case 2 – The TDS issues a stop trajectory/motion command when the force is excessive. This case is similar to CA2 Case 2 and the same constrains/requirements apply. i.e. the gradient force information, including appropriate representation of force information to surgeon.

TA3/6 Case 3 – The TDS issues a stop trajectory/motion command when the bone is in an incorrect position and the robot is in the correct position. This might lead to inaccurate reduction. This can be amended by providing relevant information through the GUI to the user to authorise such a command.

TA3/6 Case 4 - The TDS issues a stop trajectory/motion command late and when the robot is in an incorrect position. This might be due to hardware failure and can lead to patient's injury. The TDS should be aware of such situations and promptly issue stop commands. The requirement is that the process control model must be extended to include such cases.

Table IV summarises the requirements and constraints. The requirements not included in the Table IV are related to the human factors and will be discussed separately in this section.

Table II – Unsafe Control Actions for RAFS. Setup and Control Actions of the Surgeon, and Control actions of the Treatment Delivery System (TDS).

Un-Controlled Action	**Not providing causes hazard**	**Providing causes hazard**	**Early/late out-of-sequence causes hazard**	**Stopping too soon/ applying too long causes hazard**
Setup Action (SCA)				
1. Position Patient	NA	Patient's leg in a difficult configuration causing inaccurate reduction.	NA	NA
2. Position Pin	NA	Pin-Bone relative position information is incorrect	*If happens after 2D to 3D registration, alters the 3D model of the bones*	NA
3. Position Tool	NA	Pin-Bone relative position information is incorrect	*If happens after 2D to 3D registration alters the 3D model of the bones*	NA
4. Move in the OR	NA	Robot collides with OR staff	NA	NA
5. Move Polaris	Optical tools get out of view	Optical tools get out of the view	Optical tools get out of the view	NA
6. Attach Pin to Robot	Inaccurate reduction		Compromised Pin-Bone Relative Position	NA
Control Action (CA)				
1. Start Treatment	NA	NA	The setup might be incomplete, or the robot is not ready	NA
2. Stop Treatment	The Robot moves out-of-control or in an incorrect way	*The robot has not completed the bone reduction*	*The robot has not completed the bone reduction*	NA
3. Manual Override	*The Robot moves in an incorrect way*	*The commands are not providing bone reduction*	*The system has not reduced the fracture*	*The system has not reduced the fracture*
TDS Actions (TA)				
1. Set Target Position	NA	The values are wrong and the robot will move in an incorrect way	If late it will cause the robot to move in an uncontrolled way	NA
2. Start Trajectory	NA	The system is not ready	*The setup might not be complete, or the robot is not ready*	NA
3. Stop Trajectory	The Robot moves out-of-control or in an incorrect way	The robot has not completed the bone reduction	The robot has not completed the bone reduction	NA
4. Set Motor Target	NA	The values are wrong and the robot will move in an incorrect way	If late then will cause the robot to move in an uncontrolled way	NA
5. Start Motion	NA	The setup is incorrect	The setup might be incomplete, or the robot is not ready	NA
6. Stop Motion	The Robot moves out-of-control or in an incorrect way	The robot has not completed the bone reduction	The robot has not completed the bone reduction	NA
7. Force-limit Stop	The robot causes tissue damage	The robot has not completed the bone reduction	If early, the system has not completed the bone reduction. If too late, the robot causes tissue damage	NA

Note: Text in Italics indicates that the context affects the existence of hazard..

Table III – Context table for Unsafe Control Actions of the OR Staff and TDS. Accidents caused depending on the context of the control action.

Action	Emergency status	Robot State	Force Applied	Bone Position	Robot Position	If provided… …in this context		
						… any time …	… too early …	… too late …
CA2 – Stop Treatment	Emergency	(d.n.m.)	(d.n.m.)	(d.n.m.)	(d.n.m.)	No	No	**Yes (AC1)**
	No Emergency	Idle	(d.n.m.)	(d.n.m.)	(d.n.m.)	No	No	No
	No Emergency	Setup	(d.n.m.)	(d.n.m.)	(d.n.m.)	No	No	No
	No Emergency	Treatment	Excessive	(d.n.m.)	(d.n.m.)	No	**Yes (AC2)**	**Yes (AC3)**
	No Emergency	Treatment	Appropriate	Incorrect	(d.n.m.)	No	No	**Yes (AC2)**
	No Emergency	Treatment	Appropriate	Correct	Incorrect	No	No	**Yes (AC1)**
	No Emergency	Treatment	Appropriate	Correct	Correct	No	No	No
CA3 – Manual Override	Emergency	(d.n.m.)	(d.n.m.)	(d.n.m.)	(d.n.m.)	**Yes (AC1)**	-	-
	No Emergency	Idle	(d.n.m.)	(d.n.m.)	(d.n.m.)	No	No	No
	No Emergency	Setup	(d.n.m.)	(d.n.m.)	(d.n.m.)	No	No	No
	No Emergency	Treatment	Excessive	(d.n.m.)	(d.n.m.)	No	No	**Yes (AC2)**
	No Emergency	Treatment	Appropriate	Incorrect	(d.n.m.)	No	No	No
	No Emergency	Treatment	Appropriate	Correct	(d.n.m)	**Yes (AC2)**	-	-
TA2/5 - Start Trajectory/Motion	Emergency	(d.n.m.)	(d.n.m.)	(d.n.m.)	(d.n.m.)	No	No	**Yes (AC1)**
	No Emergency	Idle	(d.n.m.)	(d.n.m.)	(d.n.m.)	No	No	No
	No Emergency	Setup	(d.n.m.)	(d.n.m.)	(d.n.m.)	**Yes(AC1&AC2)**	-	-
	No Emergency	Treatment	Excessive	(d.n.m.)	(d.n.m.)	**Yes (AC3)**	-	-
	No Emergency	Treatment	Appropriate	Incorrect	(d.n.m.)	No	No	No
	No Emergency	Treatment	Appropriate	Correct	Incorrect	**Yes (AC2)**	-	-
	No Emergency	Treatment	Appropriate	Correct	Correct	No	No	No
TA3/6 - Stop Trajectory/Motion	Emergency	(d.n.m.)	(d.n.m.)	(d.n.m.)	(d.n.m.)	No	No	**Yes (AC1)**
	No Emergency	Idle	(d.n.m.)	(d.n.m.)	(d.n.m.)	No	No	No
	No Emergency	Setup	(d.n.m.)	(d.n.m.)	(d.n.m.)	No	No	No
	No Emergency	Treatment	Excessive	(d.n.m.)	(d.n.m.)	No	**Yes (AC2)**	**Yes (AC3)**
	No Emergency	Treatment	Appropriate	Incorrect	Correct	**Yes (AC2)**	No	No
	No Emergency	Treatment	Appropriate	(d.n.m.)	Incorrect	No	No	**Yes (AC1)**
	No Emergency	Treatment	Appropriate	Correct	Correct	No	No	No
TA1/4 - Set Target\ Motor	(d.n.m.)	Idle	(d.n.m.)	(d.n.m.)	(d.n.m.)	No	No	No
	(d.n.m.)	Setup	(d.n.m.)	(d.n.m.)	(d.n.m.)	No	No	No
	(d.n.m.)	Treatment	(d.n.m.)	(d.n.m.)	(d.n.m.)	**Yes(AC1&AC2)**	-	-

(d.n.m) : does not matter

Table IV – Table of Requirements as generated by the UCA case analysis.

Req. Number	Requirement Description
R1	Add emergency parameter to the TDS process model as a flag for issuing commands by the surgeon and the TDS itself
R2	Force hysteresis thresholds, i.e. triggering and releasing alerts at different force limits. Adding exceptions in manual override cases.
R3	Control of the stop treatment command based on TDS process model.
R4	System movement is assessed in terms of the bone position and not robot's end-effector position
R5	No further motion commands are issued if the reduction is complete.
R6	Clear setup mode and no movement commands in this mode.
R7	General reliability requirements for ensuring that process parameters of the TDS model are accurately reflecting robot's status.

Three major types of requirements can be seen in Table IV a) process model amendments (R1), b) use of process model parameters for contextualising commands (R3, R5, R6), and c) operational amendments (R2, R4) and overall hardware functionality (R7). The first type is intended to improve the process models and make it more comprehensive in order to cover all possible situations. The second type is intended to allow a contextualisation of decision-making. It is attempting to evaluate the state of the system and provide an informed decision for the next action. The third type includes requirements that are affecting the type of control. It provides a control approach that is using a different metric than a typical method, for example, force is not used as a fixed value but as a range, and position is controlled indirectly to achieve the optimal result. Finally, requirement R7 is intended not only as a technical reassurance but also as a mean to verify the process model. The latter is crucial for performance of the system. This is why setup actions are important.

5.2. *Mitigation of hazards from Setup Control Actions*

As seen in section 4.3, the setup actions can be compromising process parameters, potentially creating discrepancies between the actual state of the system and the model. This could be difficult to amend since the criteria of assessing a state are compromised. A possible mitigation technic could be to use redundancy structures to allow overlap of process model parameters. For example, in the case of positioning the system's parts, *Pin Position*, *Bone Position*, and *Tracker Position* can be compared to each other, and compared to internal robot information (e.g. encoder values). Similarly, force information, *Force Applied,* can be compared with the motor current data, and the robot's dynamics model to verify the state of the reduction and the interaction with soft tissues. Moreover, a comparison of position information and force information can provide an additional cross-referencing layer regarding bone-pin-robot relationship. Figure 7 gives a pictorial representation of this interaction.

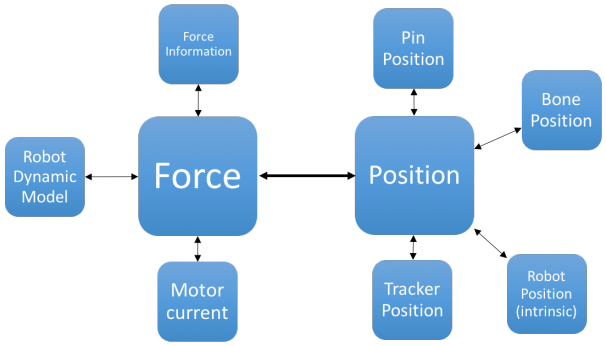


Fig. 7. Force and Position information flow validating process model parameters

In order to allow a correct interaction, the requirement R7 must be satisfied which assumes reliable hardware and the human operator's process model kept properly updated and in sync with the process model. This can be achieved by taking into account human factors when designing interactions with the system.

5.3. *Human factors for safety*

Further to the technical requirements presented in section 5.1, it is clear that the analysed cases include requirements related to the interactions of the user with the system. Human-factors related to the use of medical devices have been widely explored in the relevant literature, In [20], [21] the authors are exploring communication between the user and the system and potential consequences. Reviewing the human-factors requirements derived from the above cases bring us to the same conclusions:

- the operating surgeon should have a clear indication of the emergency state;
- appropriate representation of force information must be available to the surgeon;
- alerts that force constraints are not satisfied;
- GUI provides information to the user to authorise the stop trajectory/motion command.

All the above requirements provide clear communication of the system information. If the information that reaches the operator is not adequately clear, his/her knowledge of the process state is flawed and out-of-sync with the reality. As a result this discrepancy can lead to hazardous situations.

6. Conclusions

We have analysed safe human-robot interaction for a medical robotic system that assists the surgeon in reducing fractures. We

are using STPA as a systematic analysis tool of the control interactions. The STPA is preferred to the traditional safety approaches because it incorporates the concept of the context and the process models. We have demonstrated in the specific cases that the use of STPA enables the detection of hazardous situations when a given command can be unsafe because the context is hazardous and not necessarily the command itself. Based on these cases, we have derived requirements and constraints for the system that were not originally included in its control architecture.

Moreover, through the same cases and with the use of setup actions we demonstrated the importance of interactions between the user and the system. It has been identified that simple technical solutions will not suffice if the human-in-the-loop is not aware of the current process state. This also brings forward the importance of having up-to-date process models in order to allow for correct and safe, control interactions. To strengthen this point, a future step for this research will be testing of the safety measures in real user-case experiments with the actual system.

Acknowledgment

This is a summary of independent research funded by the National Institute for Health Research (NIHR)'s Invention for Innovation (i4i) Programme. The views expressed are those of the author(s) and not necessarily those of the NHS, the NIHR or the Department of Health.

References

[1] H. Alemzadeh, J. Raman, N. Leveson, and R. K. Iyer, "Safety Implications of Robotic Surgery: A Study of 13 Years of FDA Data on da Vinci Surgical Systems," *Univ. Illinois Coord. Sci. Lab. Tech. Report, UILU-ENG-13-2208*, 2013.

[2] M. M. Sobhani, A. G. Pipe, S. Dogramadzi, and J. G. Fennell, "Towards Model-Based Robot Behaviour Adaptation : Successful Human-Robot Collaboration in Tense and Stressful Situations," no. 238486.

[3] D. Raabe, S. Dogramadzi, and R. Atkins, "Semi-automatic percutaneous reduction of intra-articular joint fractures - An initial analysis," in *2012 IEEE International Conference on Robotics and Automation (ICRA)*, 2012, pp. 2679–2684.

[4] G. Dagnino, I. Georgilas, R. Atkins, S. Dogramadzi, and others, "Image-based robotic system for enhanced minimally invasive intra-articular fracture surgeries," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, 2016, pp. 696–701.

[5] A. Sánchez, P. Poignet, E. Dombre, A. Menciassi, and P. Dario, "A design framework for surgical robots: Example of the Araknes robot controller," *Rob. Auton. Syst.*, vol. 62, no. 9, pp. 1342–1352, 2014.

[6] L. A. Sanchez, M.-Q. Le, K. Rabenorosoa, C. Liu, N. Zemiti, P. Poignet, E. Dombre, A. Menciassi, and P. Dario, "A case study of safety in the design of surgical robots: The ARAKNES platform," in *Intelligent Autonomous Systems 12*, Springer, 2013, pp. 121–130.

[7] J. Guiochet, Q. A. Do Hoang, M. Kaâniche, and D. Powell, "Applying Existing Standards to a Medical Rehabilitation Robot : Limits and Challenges," *IEEE/RSJ Int. Conf. Intell. Robot. Syst. (IROS), Work. FW5 Saf. Human-Robot Coexistence Interact. How can Stand. Res. benefit from each other?*, p. 5, 2012.

[8] M. Jung and P. Kazanzides, "Run-time Safety Framework for Component-based Medical Robots," in *ACM/IEEE International Conference on Cyper-Physical System (Workshop of MedicalCyber Physical Systems)*, 2013, pp. 1 – 8.

[9] M. Y. Jung, R. H. Taylor, and P. Kazanzides, "Safety Design View : A Conceptual Framework for Systematic Understanding of Safety Features of Medical Robot Systems," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, 2014, pp. 1883–1888.

[10] P. Kazanzides, G. Fichtinger, G. D. Hager, A. M. Okamura, L. L. Whitcomb, and R. H. Taylor, "Surgical and interventional robotics-core concepts, technology, and design [Tutorial]," *Robot. Autom. Mag. IEEE*, vol. 15, no. 2, pp. 122–130, 2008.

[11] I. Georgilas, G. Dagnino, and S. Dogramadzi, "Human-caused hazards in medical robotics: The case of a Fracture Reduction System," in *Joint Workshop on New Technologies for Computer/Robot Assisted Surgery*, 2015.

[12] N. Leveson, "Engineering a Safer World," *J. Chem. Inf. Model.*, vol. 53, p. 555, 2011.

[13] E. Mitka and S. G. Mouroutsos, "Applying the STAMP system safety engineering methodology to the design of a domestic robot," *Int. J. Appl. Syst. Stud.*, vol. 6, no. 1, pp. 81–102, 2015.

[14] S. Procter and J. Hatcliff, "Hazard Analysis for Medical Applications," in *Formal Methods and Models for Codesign (MEMOCODE), 2014 Twelfth ACM/IEEE International Conference on*, 2014, pp. 124–133.

[15] N. Leveson, "An STPA Primer," *Version 1,* vol. 2013, no. August, 2013.

[16] G. Dagnino, I. Georgilas, P. Tarassoli, R. Atkins, and S. Dogramadzi, "Design and Real-Time Control of a Robotic System for Fracture Manipulation," in *IEEE EMBC 2015*, 2015.

[17] G. Dagnino, I. Georgilas, P. Tarassoli, R. Atkins, and S. Dogramadzi, "Vision-Based Real-Time Position Control of a Semi-automated System for Robot-Assisted Joint Fracture Surgery," *29th Congr. Comput. Assist. Radiol. Surg.*, 2015.

[18] G. Dagnino, I. Georgilas, P. Tarassoli, R. Atkins, and S.

Dogramadzi, "Intra-Operative 3D Imaging System for Robot-Assisted Fracture Manipulation," in *37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, 2015.

[19] G. Dagnino, I. Georgilas, P. Köhler, S. Morad, R. Atkins, and S. Dogramadzi, "Navigation system for robot-assisted intra-articular lower-limb fracture surgery," *Int. J. Comput. Assist. Radiol. Surg.*, pp. 1–13, 2016.

[20] C. J. Vincent, Y. Li, and A. Blandford, "Integration of human factors and ergonomics during medical device design and development: it's all about communication.," *Appl. Ergon.*, vol. 45, no. 3, pp. 413–9, May 2014.

[21] S. Sharples, J. Martin, A. Lang, M. Craven, S. O'Neill, and J. Barnett, "Medical device design in context: A model of user–device interaction and consequences," *Displays*, vol. 33, no. 4–5, pp. 221–232, Oct. 2012.

Ioannis Georgilas received his M.S. and Ph.D. degrees id Production Engineering from Democritus University of Thrace, Greece, in 2005 and 2010, respectively. From 2011 to 2013, he was a Research Associate at the Bristol Robotics Laboratory, University of the West of England, Bristol, UK. and worked on the project of Bio-inspired Unconventional Manipulation. From 2013 until 2015, he worked as a Research Fellow at Bristol Robotics Laboratory, University of the West of England, Bristol, UK, for the project of Robot Assisted Fracture Surgery. Since 2015 he is a Lecturer on Mechatronics at the University of the West of England, Bristol, UK.

Giulio Dagnino received his M.S. degree in Bioengineering from the Universita' degli Studi di Genova, Italy, and his Ph.D. degree in Medical Robotics from the Universita' degli Studi di Genova, Italy, in 2007 and 2013, respectively. From 2010 to 2013, he was at the Istituto Italiano di Tecnologia, as Doctoral Fellow. He is currently a Research Fellow at the Bristol Robotics Laboratory, University of the West of England, UK, where he is working on the NIHR funded project RAFS (Robot-Assisted Fracture Surgery). His efforts are concentrated on the development of real-time control and 3D imaging for the system.

Sanja Dogramadzi received her Ph.D. degree from the University of Newcastle, in 2001. From 2001 to 2006 she was a Post-Doctoral Researcher/ at the University of Leeds working on various projects in BioMedical Robotics and Safety of Engineering Systems. From 2006 to 2012, she has been a Senior Lecturer in Robotics at University of the West of England. Now, she holds the position of Associate Professor of Medical Robotics at the Faculty of Engineering at the University of the West of England.